```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from scipy import stats
from scipy.stats import skew, kurtosis, mode #Python libraries for inferential statistics
import seaborn as sns #This is for generating Histogram with Ker
```

```
df = pd.read_csv('hotel_books.csv')
df.head(5)
```

|   | day | clients | total_bill |
|---|---|---|---|
| **0** | 1 | 33 | 23958 |
| **1** | 2 | 25 | 26812 |
| **2** | 3 | 5 | 24871 |
| **3** | 4 | 17 | 17954 |
| **4** | 5 | 28 | 29416 |

```
df = pd.read_csv('hotel_books.csv')
df.head(5)
```

|   | day | clients | total_bill |
|---|---|---|---|
| **0** | 1 | 33 | 23958 |
| **1** | 2 | 25 | 26812 |
| **2** | 3 | 5 | 24871 |
| **3** | 4 | 17 | 17954 |
| **4** | 5 | 28 | 29416 |

```
df.dtypes
```

|   | 0 |
|---|---|
| **day** | int64 |
| **clients** | int64 |
| **total_bill** | int64 |

**dtype:** object

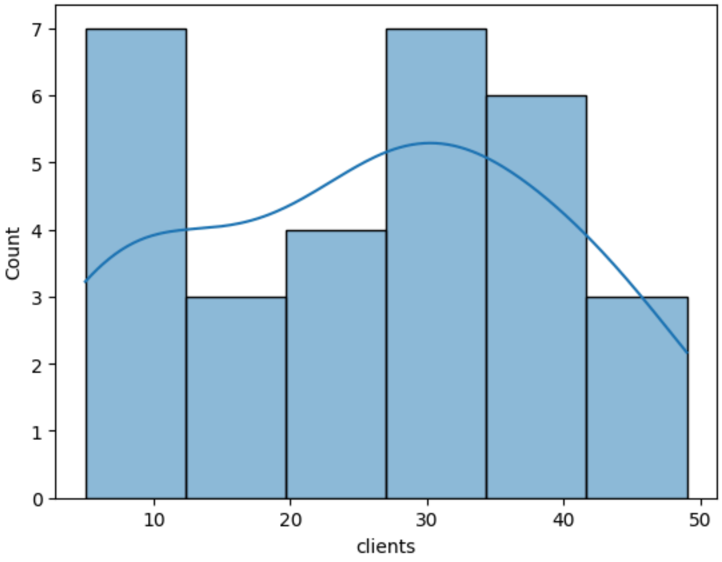
```
df.isnull().sum()
```

|   | 0 |
|---|---|
| **day** | 0 |
| **clients** | 0 |
| **total_bill** | 0 |

**dtype:** int64

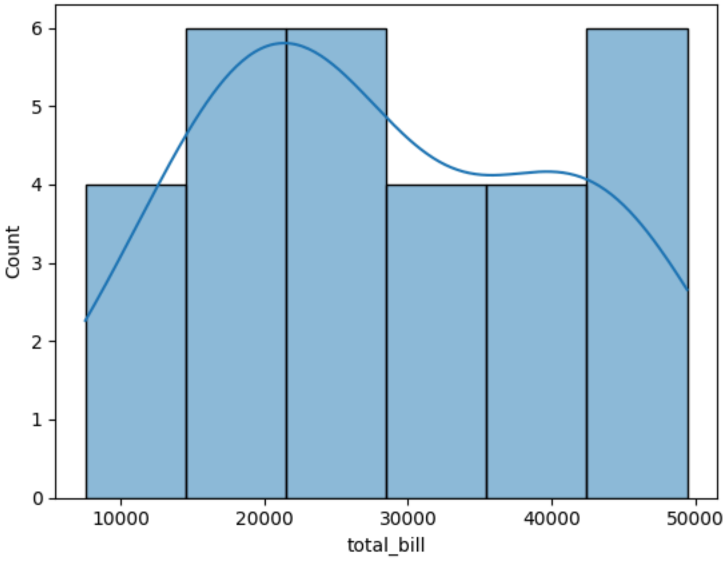
```
sns.histplot(df['clients'], kde=True)
```

```
<Axes: xlabel='clients', ylabel='Count'>
```



```
sns.histplot(df['total_bill'], kde=True)
```

```
<Axes: xlabel='total_bill', ylabel='Count'>
```



```
#compute for skewness and kurtosis for number of clients
skew1 = df['clients'].skew()
kurt1 = df['clients'].kurt()
print(f'Kurtosis for the number of hotel clients in a day:{kurt1}')
print(f'Skewness for the number of hotel clients in a day:{skew1}')
```

```
Kurtosis for the number of hotel clients in a day:-1.1388703400867874
Skewness for the number of hotel clients in a day:-0.05968808896371035
```

```
#compute for skewness and kurtosis for total number of bill
skew2 = df['total_bill'].skew()
kurt2 = df['total_bill'].kurt()
print(f'Kurtosis for the total bill collected from clients per day:{kurt2}')
print(f'Skewness for the total bill collected from clients per day:{skew2}')
```

```
Kurtosis for the total bill collected from clients per day:-1.130219880444574
Skewness for the total bill collected from clients per day:0.18976914965853053
```

```
dfibe().descr
```

```
             day    clients    total_bill
count  30.000000  30.000000     30.000000
mean   15.500000  25.666667  28344.233333
std     8.803408  13.557879  12441.769892
min     1.000000   5.000000   7534.000000
25%     8.250000  16.000000  18335.000000
50%    15.500000  28.000000  25841.500000
75%    22.750000  35.750000  39810.250000
max    30.000000  49.000000  49450.000000
```

```
stats.mode(df['clients'])
```

```
ModeResult(mode=8, count=4)
```

```
stats.mode(df['total_bill'])
```

```
ModeResult(mode=7534, count=1)
```